

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Ingo Zenz

Application No.: 10/749,957

Filed: December 30, 2003

For: **CONFIGURATION MANAGER IN
ENTERPRISE COMPUTING SYSTEM**

Examiner: Pardo, Thuy N.

Art Unit: 2815

Confirmation No.: 8586

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

The Appellant submits the following Appeal Brief pursuant to 37 C.F.R. § 41.37(c) for consideration by the Board of Patent Appeals and Interferences. The Appellant authorizes the amount of \$510.00 to cover the cost of filing the opening brief as required by 37 C.F.R. § 1.17(f) to be charged to Deposit Account No. 02-2666.

TABLE OF CONTENTS

	Page
I. REAL PARTY IN INTEREST	2
II. RELATED APPEALS AND INTERFERENCES	2
III. STATUS OF CLAIMS	2
IV. STATUS OF AMENDMENTS	2
V. SUMMARY OF THE CLAIMED SUBJECT MATTER.....	2
VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL.....	6
VII. ARGUMENT	6
A. Overview of the Cited References	6
1. E	6
2. Vahalia	7
B. Rejections Under 35 U.S.C. § 102(e)	8
C. Rejections Under 35 U.S.C. § 103(a)	19
VIII. CLAIMS APPENDIX.....	31
IX. EVIDENCE APPENDIX	35
X. RELATED PROCEEDINGS APPENDIX	36

I. REAL PARTY IN INTEREST

Ingo Zenz, the party named in the caption, transferred his rights to the subject Application through an assignment recorded on June 14, 2004 (Reel/Frame 015464/0924) in the patent application to SAP Aktiengesellschaft, of Walldorf, GERMANY. Thus, as the owner at the time the brief is being filed, SAP Aktiengesellschaft is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences that will directly affect, be directly affected by or have a bearing on the Board's decision in this Appeal.

III. STATUS OF CLAIMS

Claims 1-27 are pending and rejected in the Application. The Appellant respectfully appeals the rejection of claims 1-27.

IV. STATUS OF AMENDMENTS

No amendments were submitted after the Final Office Action, mailed on June 7, 2007.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The embodiments of the claimed subject matter provide a method, system, and computer readable storage media for management of configuration data within a cluster (*See* Specification at pg. 3, ll. 13-20).

With respect to independent claim 1, the method comprises storing a configuration for a distributed environment in a central storage of the distributed environment (*See* Specification at pg. 6, line 26 - pg. 7, line 6); and updating a portion of the configuration in the distributed environment (*See* Specification at pg. 12, line 32 - pg. 13, line 32; Fig. 5A, elements 506, 508, and 510).

Claim 2 depends from claim 1 and recites the further limitations of acquiring a lock for the portion of the configuration in a first node in the distributed environment (*See* Specification at pg. 12, ll. 13-31; Fig. 5A, element 502); modifying the portion of the configuration (*See* Specification at pg. 12, line 33 - pg. 13, ll. 1-16; Fig. 5A, element 504); invalidating a

representation of the portion of the configuration in the distributed environment (*See Specification* at pg. 14, ll. 1-16; Fig. 5B, element 536); and releasing the lock (*See Specification* at pg. 13, ll. 18-32; Fig. 5A, element 512).

Claim 3 depends from claim 2 and recites the further limitations of updating a database to reflect modifications of a portion of the configuration (*See Specification* at pg. 12, line 33 - pg. 13, line 16; Fig. 5A, element 508); and blocking reads of the configuration during the updating (*See Specification* at pg. 12, line 33 - pg. 13, line 16).

Claim 4 depends from claim 2 and recites the further limitations of notifying nodes in the distributed environment of the updated configuration data (*See Specification* at pg. 13, ll. 18-32; Fig. 5A, element 510).

Claim 7 depends from claim 2 and recites the further limitations of modifying comprises: changing a configuration object in a branch of a tree structure (*See Specification* at pg. 7, line 28 - pg. 8 line 9 and pg. 13, ll. 18-32).

Claim 8 depends from claim 2 and recites the further limitations of invalidating comprises: sending a cache invalidation event to another node in the cluster (*See Specification* at pg. 12, line 33 - pg. 13, line 16).

Claim 9 depends from claim 2 and recites the further limitations of invalidating comprises: sending a message to a plurality of Java 2 Enterprise Edition (J2EE) nodes (*See Specification* at pg. 7, ll. 8-18 and pg. 12, line 33 - pg. 13 line 16).

Claim 10 depends from claim 2 and recites the further limitations of updating further comprises: notifying registered listeners that the configuration has been changed (*See Specification* at pg. 13, ll. 18-32).

Regarding independent claim 11, the system comprises a plurality of nodes each having a instance of a configuration manager to maintain consistent storage of a configuration across the nodes without passing configuration modifications between the nodes (*See Specification* at pg. 6, line 11 - pg. 7, line 6; Fig. 1, elements 114, 116, 118, 124, 126, 128, 144, and 154); a locking server shared by the plurality of nodes to coordinate access to the configuration (*See*

Specification at pg. 5, line 33 - pg. 6, line 3; Fig. 1, element 102); and a database management system to provide an interface with a shared relational database, the database to store the configuration (*See* Specification at pg. 6, line 11 - pg. 7, line 6; Fig. 1, elements 130 and 132).

Claim 12 depends from claim 11 and recites the further limitations of the configuring manager comprises: a configuration cache (*See* Specification at pg. 8, ll. 11-31; Fig. 2, element 200); and a configuration handler (*See* Specification at pg. 9, ll. 1-10; Fig. 2, element 210).

Claim 13 depends from claim 12 and recites the further limitations of the configuration manager further comprises: a persistency handler (*See* Specification at pg. 8, ll. 11-31; Fig. 2, element 208).

Claim 14 depends from claim 11 and recites the further limitations of a configuration handler to permit access to and modification of the configuration (*See* Specification at pg. 9, ll. 1-10; Fig. 2, element 210).

Claim 15 depends from claim 11 and recites the further limitations of the configuration comprises: a plurality of persistent objects holding information about a Java 2 enterprise edition cluster (*See* Specification at pg. 7, ll. 8-18 and pg. 7, line 28 - pg. 8, line 9).

Claim 17 depends from claim 11 and recites the further limitations of the configuration manager comprises: a change event listener to notify registered components of configuration change events (*See* Specification at pg. 8, ll. 23-32; Fig. 2, element 212).

Independent claim 18 recites the limitations of a computer readable storage media containing executable computer program instructions which when executed cause a digital processing system to perform a method comprising: storing a configuration for a distributed environment in a central storage of the distributed environment (*See* Specification at pg. 6, line 26 - pg. 7, line 6); and updating a portion of the configuration in the distributed environment (*See* Specification at pg. 12, line 32 - pg. 13, line 32; Fig. 5A, elements 506, 508, and 510).

Claim 20 depends from claim 19 and recites the further limitations of updating comprises: updating a database to reflect modifications of a portion of the configuration (*See*

Specification at pg. 12, line 33 - pg. 13, line 16; Fig. 5A, element 508); and blocking reads of the configuration during the updating (*See* Specification at pg. 12, line 33 - pg. 13, line 16).

Claim 21 depends from claim 19 and recites the further limitations of updating comprises: notifying node in the distributed environment of the current configuration data (*See* Specification at pg. 13, ll. 18-32; Fig. 5A, element 510).

Claim 23 depends from claim 19 and recites the further limitations of invalidating comprises: sending a cache invalidation event to another node in the cluster (*See* Specification at pg. 12, line 33 - pg. 13, line 16).

Claim 24 depends from claim 19 and recites the further limitations of updating comprises: notifying registered listeners that the configuration has been changed (*See* Specification at pg. 13, ll. 18-32).

Independent claim 25 recites the limitations of a system comprising: means for maintaining consistent storage of configuration information in a distributed environment (*See* Specification at pg. 6, line 11 - pg. 7, line 6; Fig. 1, elements 130 and 132); means for controlling access to the configuration information (*See* Specification at pg. 5, line 33 - pg. 6, line 3; Fig. 1, element 102); and means for interfacing with a relational database system to provide persistent storage of the configuration information (*See* Specification at pg. 6, line 11 - pg. 7, line 6; Fig. 1, elements 114, 116, 118, 124, 126, 128, 144, and 154).

Claim 26 depends from claim 25 and recites the further limitations of the configuration information comprises: a plurality of persistent objects holding information about a Java 2 Enterprise Edition cluster (*See* Specification at pg. 7, ll. 8-18 and pg. 7, line 28 - pg. 8, line 9).

Claim 27 depends from claim 25 and recites the further limitations of the means for maintaining comprises: a configuration cache resident in each node of the distributed environment (*See* Specification at pg. 8, ll. 11-31; Fig. 2, element 200); and a configuration handler resident in each node of the distributed environment (*See* Specification at pg. 9, ll. 1-10; Fig. 2, element 210).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-6, 10, 11, 17-22, 24, and 25 stand rejected under 35 U.S.C. § 102(e) as being anticipated by E *et al.* (US 2004/0019639) (“E”). Claims 7, 8, 12-14, 23, and 27 stand rejected under 35 U.S.C. § 103(a) as being obvious over E, in view of Vahalia *et al.* (US 2005/0251500) (“Vahalia”). Claims 9, 15, and 26 stand rejected under 35 U.S.C. § 103(a) as being obvious over E, in view of the Appellant’s alleged admission of prior art. Claim 16 stands rejected under 35 U.S.C. § 103(a) as being obvious over E, in view of Vahalia, and further in view of the Appellant’s alleged admission of prior art.

All of the claims do not stand or fall together. The basis for the separate patentability of the claims is set forth below.

VII. ARGUMENT

A. Overview of the Cited References

1. E

E describes a distributed data system in which access to portions of distributed data may be restricted through the use of a lock mechanism (*See E* at ¶ 0014). Distributed data refers, for example, to “components of session data representing client sessions for clients of a multi-tiered enterprise application” (*See Id.* at ¶ 0012). Locks can restrict access to portions of distributed data by denying access to processes that do not hold the lock. Threads of a process can request a lock and use it to access the portion of distributed data corresponding to the lock. These requests may be buffered so as to limit the number of provided locks at any given time (*See Id.* at ¶ 0014-0016).

Another aspect of E involves the release of locks. When a thread releases a lock, other threads of the process are blocked from access to that portion of distributed data until the process acquires the lock again (*See Id.* at ¶ 0067). A lock may be released according to various circumstances. For instance, a portion of distributed data representing “session data for a client session of an enterprise server” may correspond to a predetermined session time. When the session has expired, locks for the portions of distributed data representing the sessions may be released as required by the processes (*See Id.* at ¶ 0018). In another example, a count can track

how many threads still require access to a portion of data. When a thread requires access, the count would increment, and when a thread completes its access, the count would decrement. A count that has decremented to a value of zero indicates that no more threads of the process require access to that portion of data. The last thread of a process that accesses the portion of data may release the lock for the process (*See Id.* at ¶ 0017 and 0058).

E does not disclose the storage and modification of a configuration for a distributed environment. E also does not disclose a listener, a registered listener, invalidating a representation of a configuration, notifying nodes of changes in a configuration, notifying registered listeners of changes in a configuration, a configuration manager that maintains consistent storage of a configuration across nodes, a configuration cache, a configuration handler, and a persistency handler.

2. Vahalia

Vahalia discloses accessing a file in a data network containing a high-speed data link that bypasses the server and couples the data storage or server to the client (*See Vahalia at ¶ 0015-0017*). The bypass data paths may be internal to a cached disk array in the server and provide for different access methods in the same file server network so as not to overload the network (*See Id.* at ¶ 0055 and 0063). When a client wishes to access a file within a local directory tree (indicated by a “tree identifier”), the client obtains a lock and corresponding metadata from the file server in order to read or write to the file. The metadata indicates the locations for the data in storage. The client uses the metadata to produce a command and then uses the command to read or write to the file. If the client writes to the file, the client may also choose to modify the metadata and return the modified metadata to the file server (*See Id.* at ¶ 0014 and Abstract). The reason the client needs to request permission to access a file by obtaining a lock is to protect against potential issues of data inconsistency. For instance, if a write operation is performed without this precaution, then data in the cache may be inconsistent with data in another cache or data as it currently stands (*See Id.* at ¶ 0051 and 0054).

Vahalia does not disclose the storage and modification of a configuration for a distributed environment. Vahalia also does not disclose a listener, a registered listener, invalidating a representation of a configuration, notifying nodes of changes in a configuration, notifying

registered listeners of changes in a configuration, a configuration manager that maintains consistent storage of a configuration across nodes, a configuration cache, a configuration handler, and a persistency handler.

B. Rejections Under 35 U.S.C. § 102(e)

Claims 1-6, 10, 11, 17-22, 24, and 25 stand rejected under 35 U.S.C. § 102(e) as being anticipated by E. It is requested that this rejection be reversed for at least the following reason. E does not expressly or inherently describe each and every element of the claims.

For a prior art reference to anticipate, 35 U.S.C. §102 requires that “**each and every element** as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.”

In re Robertson, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950 (Fed. Cir. 1999) (*quoting Verdegaal Bros., Inc. v. Union Oil Co.*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987)) (emphasis added).

1. Independent Claims 1 and 18

a) Independent Claims 1 and 18 are not anticipated at least because E fails to describe a configuration stored in a distributed environment

In particular, Independent Claims 1 and 18 recite: “storing a configuration for a distributed environment in a central storage of the distributed environment; and updating a portion of the configuration in the distributed environment” (emphasis added). E fails to expressly or inherently describe these aspects.

The Examiner contends that E describes these aspects at the Abstract, Figure 2, and ¶ 0035, 0060-0062, 0071, 0077, and 0103 (See Final Office Action mailed June 7, 2007 at pgs. 8 and 9). The Appellant respectfully avers to the contrary because at the indicated passages, E discloses storing **distributed data**, not “storing a configuration for a distributed environment” as claimed. Distributed data is not equivalent to a configuration for a distributed environment. As set forth by MPEP § 2111, claims should be interpreted through their broadest reasonable construction in light of the Specification as viewed by one of ordinary skill in the art.

The Patent and Trademark Office ("PTO") determines the scope of claims in patent applications not solely on the basis of the claim language, but *upon giving claims their broadest reasonable construction "in light of the specification as it would be interpreted by one of ordinary skill in the art."* *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364[, 70 USPQ2d 1827] (Fed. Cir. 2004). Indeed, the rules of the PTO require that application claims must "conform to the invention as set forth in the remainder of the specification and the terms and phrases used in the claims must find clear support or antecedent basis in the description so that the meaning of the terms in the claims may be ascertainable by reference to the description." 37 CFR 1.75(d)(1).

415 F.3d at 1316, 75 USPQ2d at 1329. *See also In re Hyatt*, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000). (emphasis added) One of ordinary skill in the art interpreting the claim in view of the Specification would not consider "a configuration for a distributed environment" to be disclosed by the "distributed data" of E.

The Examiner acknowledges that a configuration is "a set of persistent objects" (*See* Final Office Action mailed June 7, 2007 at pg. 8). However, the Examiner then asserts that "it is also an information (the configuration is a configuration of information, see the abstract), data and/or program code (see 0009, 0027-0028 of specification), messages (0023), or a tree structure, a set of name value pairs, set of files and/or a set of sub-configuration objects (0030 of specification)" (*See* Final Office Action mailed June 7, 2007 at pgs. 8 and 9). The Appellant respectfully disagrees with the Examiner's interpretation of the Abstract and cited paragraphs of the Specification.

One of ordinary skill in the art would interpret "a configuration for a distributed environment" as the structure established for a distributed environment. The Specification is consistent with this interpretation. The Abstract does not say that "the configuration is a configuration of information" as the Examiner contends, but rather states that "[e]ach server includes a configuration manager to facilitate changes to configuration information within the cluster." The Examiner's interpretation of the sentence in the Abstract fails to recognize the part of speech of the term as used in context. Where the sentence refers to "configuration

information,” “configuration” is an adjective that modifies the noun “information.” In the claims, “configuration” is a noun used alone (*See e.g.*, Independent Claims 1 and 18) and thus cannot equate to mere “distributed data.”

In reference to the Examiner’s cited paragraphs of the Specification, it should not be surprising that a configuration would provide information or data about the configuration (*i.e.*, configuration information and configuration data). But, this does not make just any type of data or information a configuration as the Examiner appears to assert. The fact that the configuration may specify, for example, binaries required for boot does not render those binaries part of the configuration. While it is true that configuration objects may include name value pairs, sets of files or sub-configuration objects, not all name value pairs or sets of files are necessarily configuration objects. Thus it does not stand to reason that, because a set of files may be stored in the distributed store of E, E satisfies the limitations of storing the configurations as claimed.

In addition, ¶ 0023 of the Specification does not indicate that messages are a configuration, but that “cache configuration messages” can be broadcast to servers. Nowhere in the paragraph is a message equated with a configuration. Moreover, referring to the Examiner’s assertion that a configuration is a tree, a tree merely refers to a hierarchical arrangement into which the set of configuration objects may be arranged. A tree is not a configuration; it is a set of objects which may be organized as a tree which is the configuration.

The Examiner characterized primary data 112 as the configuration (*See* Final Office Action mailed June 7, 2007 at pgs. 8 and 9). However, there is no indication in E that the primary data represents “a configuration for a distributed environment” (emphasis added). E does not disclose in express or inherent terms a “configuration for a distributed environment,” but rather discloses data (*See e.g.* E at ¶ 0012, “session data”). In ¶ 0035, E reflects that data stored in the distributed environment may include objects, strings, integers, Booleans, characters, or any other type of computer-representable data. However, this is merely reflective of the type of data and does not indicate that the data represents “a configuration for a distributed environment” (emphasis added).

In other words, data may be **configured** in a certain manner, but that does not equate to a configuration which itself is the data. For example, ¶ 0012 of E sets forth that “distributed data

may include components of session data representing client sessions for clients of a multi-tiered enterprise application.” Thus, E does not describe “storing a *configuration for a distributed environment* in a central storage of the distributed environment” (emphasis added) let alone “updating a portion of the *configuration* in the distributed environment” (emphasis added), as recited in independent claims 1 and 18.

In view of at least the foregoing, it is readily apparent that E does not describe each and every element of Independent Claims 1 and 18. Accordingly, the Appellant respectfully requests that the rejections of Independent Claims 1 and 18 be reversed.

2. Claims 2, 5, 6, 19, and 22

a) Claims 5 and 6 depend from Claim 2 and Claim 22 depends from Claim 19; Claims 2 and 19 depend from patentable Base Claims 1 and 18, respectively

Claims 5 and 6 depend from Claim 2 and Claim 22 depends from Claim 19. Claims 2 and 19 depend from Base Claims 1 and 18, respectively, and are directed toward allowable subject matter for at least the reasons mentioned in connection with Claims 1 and 18.

Accordingly, for at least these reasons, the Appellant respectfully requests that the § 102(e) rejection of Claims 2, 5, 6, 19, and 22 be overturned. Further, the Appellant believes that these claims are separately patentable for the reasons below.

b) Claims 2 and 19 are not anticipated at least because E fails to describe modifying or invalidating a portion of the configuration

Claims 2 and 19 recite: “*modifying* the portion of the *configuration*; *invalidating* a representation of the portion of the *configuration* in the distributed environment” (emphasis added). E fails to describe these additional aspects of the claims.

The Examiner contends that E discloses these aspects at Figure 6 and ¶ 0042, 0060, 0062, 0071, 0073, and 0103 (See Final Office Action mailed June 7, 2007 at pgs. 3 and 9). The Appellant respectfully avers to the contrary.

In Figure 6 and paragraphs 0042, 0060, 0062, 0071, 0073, and 0103, E teaches locking and modifying **distributed data**, but fails to teach modifying a portion of a **configuration** or invalidating a **representation** of a portion of a **configuration**. The cited reference discloses a thread of a process that acquires a lock for a portion of distributed data and uses it to access the portion of distributed data. All threads of a process access the data by turn, until the last thread accesses the data. This last thread releases the lock to the distributed store so that the process would have to acquire a lock once again to access the distributed data. As described previously, since there is no mention in E of storing a configuration, it follows that no related actions corresponding to the configuration would be taught in E. Further, because all of the threads access the data directly in turn, there is no need to create any representation of the data. Thus, there is no representation to invalidate.

Therefore, the Examiner has failed to establish that E teaches each of the elements of claims 2 and 19. Accordingly, these claims are separately patentable and the Appellant respectfully requests that the anticipation rejection be reversed.

3. Claims 3 and 20

a) Claims 3 and 20 depend from patentable Claims 2 and 19, respectively

Claims 3 and 20 depend from Claims 2 and 19, respectively, and are directed toward allowable subject matter for at least the reasons mentioned in connection with Claims 2 and 19. Accordingly, for at least these reasons, the Appellant respectfully requests that the § 102(e) rejection of Claims 3 and 20 be overturned. Further, the Appellant believes that these claims are separately patentable for the reasons below.

b) Claims 3 and 20 are not anticipated at least because E fails to describe blocking reads of the configuration

Claims 3 and 20 recite: “updating a database to reflect modifications of a portion of the configuration; and blocking reads of the configuration during the updating.” E does not expressly or inherently describe these additional aspects.

The Examiner contends that E teaches these aspects at Figures 3C and 5B, as well as ¶ 0042, 0060-0062, 0071, 0077, and 0103 (See Final Office Action mailed June 7, 2007 at pgs. 3 and 9). The Appellant respectfully disagrees.

In the indicated Figures and paragraphs, E discloses denying access when data is **locked**. The passages further explain that locks may be released when appropriate. Claims 3 and 20 recite blocking reads of the configuration during the **updating**—where the updating is a limitation separate from the lock of claim 2. The updating necessarily occurs **after** the modifying aspect of claim 2 because the update **reflects** the modifications. In addition, since E is silent regarding a configuration, it follows that E is also silent regarding updating a database regarding the configuration and blocking reads of the configuration during the updating.

Therefore, the Examiner has failed to establish that E teaches each of the elements of claims 3 and 20. Accordingly, these claims are separately patentable and the Appellant respectfully requests that the anticipation rejection be reversed.

4. Claims 4 and 21

a) Claims 4 and 21 depend from patentable Claims 2 and 19, respectively

Claims 4 and 21 depend from Claims 2 and 19, respectively, and are directed toward allowable subject matter for at least the reasons mentioned in connection with Claims 2 and 19. Accordingly, for at least these reasons, the Appellant respectfully requests that the § 102(e) rejection of Claims 4 and 21 be overturned. Further, the Appellant believes that these claims are separately patentable for the reasons below.

b) Claims 4 and 21 are not anticipated at least because E fails to describe notifying nodes in the distributed environment of the updated configuration data

Claims 4 and 21 recite: “notifying nodes in the distributed environment of the updated configuration data.” E does not expressly or inherently describe this additional aspect.

The Examiner contends that E teaches this aspect at ¶ 0048-0052 (*See* Final Office Action mailed June 7, 2007 at pg. 9). The Appellant respectfully disagrees with this contention.

In ¶ 0048-0052, E discusses a thread within the local environment notifying a “local data manager” when it requires access and also when it has completed access. The local data manager can keep a “count mechanism” that increments whenever a thread requests access to a portion of distributed data and that decrements when a thread completes access of the portion of distributed data (and therefore no longer requires access to the data). An indication that the count is zero could mean that “no threads of process 106 require access to the distributed data portion” and “may trigger an event that may notify first thread 210 or the local data manager that there are no more threads requiring access to the distributed data portion.” When that is the case, the thread that just completed access may “release the lock for the portion of primary data 112 for process 106” (*See* E at ¶ 0049 and 0050).

Because E is silent regarding configuration data, there is no indication that configuration data has been updated. In addition, E does not disclose “notifying nodes in the distributed environment of the updated configuration data,” but rather describes notifying a “local data manager” when a thread requests access and when the thread completes its access to the data. A “local data manager” that keeps track of a count of threads is not equivalent to nodes that are notified of updated configuration data. Further, the notification in E simply provides enough information to track a count of threads, but does not go as far as “notifying … of the updated configuration data.”

Thus, the Examiner has failed to establish that E teaches each of the elements of claims 4 and 21. Accordingly, these claims are separately patentable and the Appellant respectfully requests that the anticipation rejection be reversed.

5. Claims 10 and 24

a) Claims 10 and 24 depend from patentable Claims 2 and 19, respectively

Claims 10 and 24 depend from Claims 2 and 19, respectively, and are directed toward allowable subject matter for at least the reasons mentioned in connection with Claims 2 and 19.

Accordingly, for at least these reasons, the Appellant respectfully requests that the § 102(e) rejection of Claims 10 and 24 be overturned. Further, the Appellant believes that these claims are separately patentable for the reasons below.

b) Claims 10 and 24 are not anticipated at least because E fails to describe notifying registered listeners that the configuration has been changed

Claims 10 and 24 recite: “notifying registered listeners that the configuration has been changed.” E does not expressly or inherently describe this additional aspect.

The Examiner contends that E teaches this aspect at ¶ 0048 and 0049 and explains that the local data manager is notified of threads requiring access to distributed data (*See* Final Office Action mailed June 7, 2007 at pg. 4). The Appellant respectfully avers to the contrary.

As explained above, E discloses the storage of distributed data, but does not disclose the storage of a configuration. Thus, E logically fails to disclose a changed configuration and notifying registered listeners of such event. In addition, since ¶ 0048 and 0049 discusses notifying a local data manager of threads that request access to data and that have completed access to the data, the Examiner appears to be equating E’s local data manager with Claims 10 and 24’s registered listener. However, a local data manager is not a **listener**, let alone a **registered listener**. In addition, E describes a notification regarding the status of certain threads that will access or have already accessed distributed data, but does not describe a notification that a configuration has been changed. In other words, E’s notification indicates whether a thread does or does not require access to the data, but such notification is silent with respect to a notification that a configuration has been **changed**. In Claims 10 and 24, when a configuration has been **changed**, a registered listener is notified (*See* E at ¶ 0048, “[i]f a thread of process 106 requests access to the distributed data portion (including first thread 210), the count may be incremented.”). In E, when a thread **requires access** to distributed data, a local data manager increments a count of threads (*See* Id., “[i]f the thread no longer requires access to the distributed data portion (e.g. when a thread completes its access of the data), the count may be decremented. For example, the thread may have finished executing.”). Also in E, when a thread that previously

required access has already completed access, the local data manager decrements a count of threads.

Therefore, the Examiner has failed to establish that E teaches each of the elements of claims 10 and 24. Accordingly, these claims are separately patentable and the Appellant respectfully requests that the anticipation rejection be reversed.

6. Independent Claim 11

a) Independent Claim 11 is not anticipated at least because E fails to describe a plurality of nodes each having a instance of a configuration manager to maintain consistent storage of a configuration across the nodes without passing configuration modifications between the nodes

Independent claim 11 recites: “a plurality of nodes each having a instance of a configuration manager to maintain consistent storage of a configuration across the nodes without passing configuration modifications between the nodes.” E does not expressly or inherently describe this aspect.

The Examiner contends that E teaches this aspect at Figure 1 and ¶ 0032, 0034, 0035, 0040-0045, and 0053 and notes that an application server is equivalent to the recited configuration manager (*See* Final Office Action mailed June 7, 2007 at pg. 10). The Appellant respectfully avers to the contrary.

As explained above, E discloses the storage of distributed data, but does not disclose the storage of a configuration. The Examiner’s citation of Figure 1 and ¶ 0032, 0034, 0035, 0040-0045, and 0053 describes an application server that involves distributed data. The disclosure of an application server does not expressly or inherently describe a configuration manager that involves the maintenance of a configuration because distributed data is not equivalent to a configuration. The application server of E manages updates of distributed data to a “distributed store” (*See* E at ¶ 0035). This is done “to synchronize the primary data with the local data” (*See Id.*). However such synchronization does not achieve the aspect of Independent Claim 11, which is “to maintain consistent storage of a configuration across the *nodes* without passing configuration modifications between the nodes” (emphasis added). Changes across more than

one node in a cluster would not be realized by a sole update to the “distributed store” as described in E—there is no indication in the cited reference that a change in the “distributed store” would affect other nodes.

Therefore, the Examiner has failed to establish that E teaches each of the elements of claim 11. Accordingly, this claim is separately patentable and the Appellant respectfully requests that the anticipation rejection be reversed.

7. **Claim 17**

a) Claim 17 depends from patentable Base Claim 11

Claim 17 depends from Base Claim 11 and is directed towards allowable subject matter for at least the reasons mentioned in connection with Claim 11. Accordingly, for at least these reasons, the Appellant respectfully requests that the § 102(e) rejection of Claim 17 be overturned. Further, the Appellant believes that this claim is separately patentable for the reasons below.

b) Claim 17 is not anticipated at least because E fails to describe a change event listener to notify registered components of configuration change events

Claim 17 recites: “a change event listener to notify registered components of configuration change events.” E does not expressly or inherently describe this additional aspect.

The Examiner contends that E teaches this aspect at Figure 1 and ¶ 0035 (See Final Office Action mailed June 7, 2007 at pg. 10). The Examiner states that an application server is equivalent to a change event listener (See Id.). The Appellant respectfully avers to the contrary.

In Figure 1 and ¶ 0035 of E, the application server “sends updates of distributed data to distributed data store 110 in response to an event such as a modification of one or more attributes of the local data 108 and/or as routine maintenance to synchronize the primary data with the local data 108.” While E’s application server notifies a “distributed data store” of “updates of distributed data,” Claim 17 recites a change event listener that notifies “registered components” of “configuration change events.”

As similarly explained above, E discloses the storage of distributed data, but does not disclose the storage of a configuration. Thus, E also fails to teach configuration change events, let alone a change event listener to notify the registered components of the configuration change events. In addition, there is no indication in E that any other component besides a “distributed data store” would receive any type of notification; thus E fails to teach a listener that notifies “components” as recited in plural form. Furthermore, there is also no indication in E that a distributed store may be **registered** as a registered component.

Therefore, the Examiner has failed to establish that E teaches each of the elements of claim 17. Accordingly, this claim is separately patentable and the Appellant respectfully requests that the anticipation rejection be reversed.

8. **Independent Claim 25**

a) Independent Claim 25 is not anticipated at least because E fails to describe an equivalent structure as governed by 35 U.S.C. § 112, ¶ 6

Independent Claim 25 recites: “means for maintaining consistent storage of configuration information in a distributed environment; means for controlling access to the configuration information; and means for interfacing with a relational database system to provide persistent storage of the configuration information.” E does not expressly or inherently describe these aspects.

The Examiner contends that the teaching in E corresponds to the arguments set forth for Claims 1-6. However, Independent Claim 25 is written in means plus function format and is governed by 35 U.S.C. § 112, ¶ 6 and therefore would require the equivalent structure (*See MPEP § 2183*) to that which is disclosed in the Appellant’s Specification (*See Specification at Fig. 1*). As described above, E teaches storage of distributed data, but fails to teach storage of configuration information. Therefore, E logically fails to teach the structures involving the configuration information (*i.e.*, “means for **maintaining** consistent storage of configuration information in a distributed environment; means for **controlling** access to the configuration information; and means for **interfacing** with a relational database system to provide persistent storage of the configuration information” (emphasis added)).

While E provides for a management and storage system for distributed data, E's system would not be able to handle maintaining, controlling, and interfacing configuration information in place of distributed data. Managing a consistent and persistent storage of configuration information calls for a broader approach, as multiple instances each containing multiple nodes have to be maintained in an organized and holistic manner, as illustrated in Figure 1.

Therefore, the Examiner has failed to establish that E teaches each of the elements of claim 25. Accordingly, this claim is separately patentable and the Appellant respectfully requests that the anticipation rejection be reversed.

For the foregoing reasons, it is respectfully requested that the rejection of Claims 1-6, 10, 11, 17-22, 24, and 25 under 35 U.S.C. § 102(e) be reversed.

C. Rejections Under 35 U.S.C. § 103(a)

Claims 7, 8, 12-14, 23, and 27 stand rejected under 35 U.S.C. § 103(a) as being obvious over E, in view of Vahalia. This rejection should be withdrawn for at least the following reason. To establish a *prima facie* case of obviousness, the Examiner must show that the cited references, combined, teach or suggest each of the elements of the claims.

1. Claim 7

a) Claim 7 depends from patentable Claim 2

Claim 7 depends from Claim 2 and is directed toward allowable subject matter for at least the reasons mentioned in connection with Claim 2. The Examiner has not relied upon and the Appellant is unable to discern any part of Vahalia that cures the aforementioned deficiencies regarding the claim limitations of Claim 2. Accordingly, for at least these reasons, the Appellant respectfully requests that the § 103(a) rejection of Claim 7 be overturned. Further, the Appellant believes that this claim is separately patentable for the reasons below.

b) Claim 7 is not obvious at least because E and Vahalia, alone or in combination, fail to describe changing a configuration object in a branch of a tree structure

Claim 7 recites: “changing a configuration object in a branch of a tree structure.” E and Vahalia, alone or in combination, do not teach or suggest this additional aspect.

The Examiner concedes that E does not teach this aspect, but contends that Vahalia teaches this aspect at Figs. 9, 10, 13-15, and 22 and ¶ 0105, 0111, 0167, and 0168 (See Final Office Action mailed June 7, 2007 at pgs. 5 and 11). The Appellant respectfully avers to the contrary.

In Figs. 9, 10, 13-15, and 22 and ¶ 0105, 0111, 0167, and 0168, Vahalia describes the processing of SMB messages. The mention of the term “tree” refers to a “tree identification field” that is set in an SMB header to indicate that a client may access the requested file system (See Vahalia at ¶ 0092). The “tree identification field” indicates the location of the file in the local directory tree (See Id. at ¶ 0014).

However, there is no teaching in E or Vahalia of object oriented data storage, and therefore the cited references, combined, do not teach or suggest “changing a **configuration object**” (emphasis added). E discloses “distributed data” and Vahalia discloses “files,” but neither cited reference teaches a configuration **object**.

In addition, while the Examiner concedes that there is no teaching of a tree structure in E, the Examiner appears to rely on the directory file system of Vahalia as teaching the tree structure (See Final Office Action mailed June 7, 2007 at pgs. 5 and 11). However, the local directory structure of Vahalia stores files (See Vahalia at Abstract and ¶ 0092) and does not store **configuration objects**. Thus, Vahalia does not teach “changing a **configuration object** in a branch of a tree structure” (emphasis added), as recited in claim 7.

Therefore, the Examiner has failed to establish that E and Vahalia, alone or in combination, teach all the limitations of claim 7. Accordingly, this claim is separately patentable and the Appellant respectfully requests that the obviousness rejection be reversed.

2. Claims 8 and 23

a) Claims 8 and 23 depend from patentable Claims 2 and 19, respectively

Claims 8 and 23 depend from Claims 2 and 19, respectively, and are directed toward allowable subject matter for at least the reasons mentioned in connection with Claims 2 and 19. The Examiner has not relied upon and the Appellant is unable to discern any part of Vahalia that cures the aforementioned deficiencies regarding the claim limitations of Claims 2 and 19. Accordingly, for at least these reasons, the Appellant respectfully requests that the § 103(a) rejection of Claims 8 and 23 be overturned. Further, the Appellant believes that these claims are separately patentable for the reasons below.

b) E is not Properly Combined with Vahalia

The Appellant does not believe that the combination of E and Vahalia is proper. To establish a *prima facie* case of obviousness, there must be a motivation to combine the teachings of E and Vahalia found either in the references themselves or as viewed by one of ordinary skill in the art.

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, *there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings*. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations.

(See MPEP § 2143) (emphasis added) In addition, the combination of E and Vahalia cannot change the principle of operation.

If the proposed modification or combination of the prior art would *change the principle of operation* of the prior art invention being modified, then the teachings of the references are *not sufficient* to render the claims *prima facie* obvious. *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959)

(See MPEP § 2143.01(VI)) (emphasis added) The Examiner does not indicate why it would be obvious to one of ordinary skill in the art to combine E and Vahalia in the rejection of claims 8 and 23 of the Final Office Action mailed June 7, 2007. Instead, the Examiner cites the sections of E and Vahalia that allegedly teach the claimed aspects without further mention of how the two references may be properly combined (See Final Office Action mailed June 7, 2007 at pg. 6).

The Appellant respectfully submits that E and Vahalia are not properly combinable because their teachings are technologically incompatible. Claims 8 and 23 recite: “sending a cache invalidation event to another node in the cluster.” The Examiner contends that “E further teaches distributed sessions may be distributed among multiple servers, for example in a cluster [0008; 0035]” and that “Vahalia further teaches sending a cache invalidation event to another node in the cluster [0167; 0127-0133]” (See Final Office Action mailed June 7, 2007 at pg. 6). However, to modify E’s server structure regarding implementation “across one or more of the same computer systems” (See E at ¶ 0035) to correspond with Vahalia’s directory file system would completely change E’s principle of operation as a server/network structure. Although Vahalia teaches invalidating a cache if the inode is changed (See Vahalia at ¶ 0167), this inode is not the same as a node in a cluster. An inode is metadata for a file that includes information about the file (See Id. at ¶ 0161 and 0165). Even though inode contains the word node, an inode is specific to file management and is unrelated to the cluster nodes recited in the claims. Thus to alter E’s implementation to correspond to Vahalia’s file structure would be to change E’s principle of operation, which is improper according to MPEP § 2143.01(VI).

The Appellant requests the Examiner to update the obviousness rejection in view of the cited guidelines with respect to *KSR International Co. v. Teleflex Inc.*, No. 04-1350 (Supreme Court, April 30, 2007).

Therefore, the Examiner has failed to establish that E and Vahalia are properly combinable. The Appellant respectfully requests that the obviousness rejection be reversed.

c) Claims 8 and 23 is not obvious at least because E and Vahalia, alone or in combination, fail to describe sending a cache invalidation event to another node in the cluster

Claims 8 and 23 recite: “sending a cache invalidation event to another node in the cluster.” E and Vahalia, alone or in combination, do not teach or suggest this additional aspect.

The Examiner indicates that Vahalia teaches “sending a cache invalidation event to another node in the cluster” at ¶ 0127-0133 and 0167 (*See* Final Office Action mailed June 7, 2007 at pg. 6). The Appellant respectfully avers to the contrary.

In ¶ 0127-0133 and 0167, Vahalia describes locks used to protect files from access by data movers without proper permission from the owners of the files (*See* Vahalia at ¶ 0127-0133). File data may be cached and caches may be “invalidated if the inode is changed” (*See* Id. at ¶ 0167). The inode is metadata corresponding to a file that contains information about the file (*See* Id. at ¶ 0161 and 0165). Therefore, if the information about the file contained in the inode changes, the cache containing the file data may be invalidated. However, the cited reference does not appear to teach “sending a cache invalidation event to another node in the cluster.” Vahalia falls silent regarding sending any information “to another node in the cluster” because the inode is not a node in a cluster. When a cache is invalidated, there is no indication that this invalidation would subsequently be sent to any nodes.

Furthermore, the Appellant is not asserting a claim reciting cache invalidation events generally, but cache invalidation events used in a context of invalidating a representation of the portion of the configuration in the distributed environment (*See* Claims 2 and 19 “modifying the portion of the configuration; invalidating a representation of the portion of the configuration in the distributed environment”). As such, the normal use of cache invalidation of file data as disclosed in Vahalia fails to teach or suggest “sending a cache invalidation event to another node in the cluster.”

Therefore, the Examiner has failed to establish that E and Vahalia, alone or in combination, teach all the limitations of claims 8 and 23. Accordingly, these claims are

separately patentable and the Appellant respectfully requests that the obviousness rejection be reversed.

3. **Claim 12**

a) Claim 12 depends from patentable Base Claim 11

Claim 12 depends from patentable Base Claim 11. Thus, Claim 12 is directed toward allowable subject matter for at least the reasons mentioned in connection with Independent Claim 11. The Examiner has not relied upon and the Appellant is unable to discern any part of Vahalia that cures the aforementioned deficiencies regarding the claim limitations of Independent Claims 11. Accordingly, for at least these reasons, the Appellant respectfully requests that the § 103(a) rejection of Claim 12 be overturned. Further, the Appellant believes that this claim is separately patentable for the reasons below.

b) Claim 12 is not obvious at least because E and Vahalia, alone or in combination, fail to teach or suggest a configuration cache and a configuration handler

Claim 12 recites: “a configuration cache; and a configuration handler.” E and Vahalia, alone or in combination, do not teach or suggest these additional aspects.

The Examiner contends that Vahalia discloses a configuration cache and a configuration handler at ¶ 0123-0125 and Fig. 18, elements 330, 323, and 324 (*See* Final Office Action mailed June 7, 2007 at pg. 6). The Appellant respectfully avers to the contrary.

In Figure 18, Vahalia illustrates cached data, but fails to describe a cache specific to storing a configuration. Elements 330, 323, and 324 of Figure 18 identify a cached disk array, a data mover #2, and a data mover #1, respectively. The data movers represent the two possible paths connecting clients to the files. As described above, the Appellant submits that distributed data is not equivalent to a configuration for a distributed environment. In ¶ 0123-0125, Vahalia discloses “stream handler routines” that manage the transmission of message packets between a data mover and a network client, as well as between two data movers. The cited reference’s “stream handler routines” are not concerned with a cache that contains a configuration or a

handler that manages a configuration because the message packets do not appear to be relevant to a configuration (*See e.g.*, Vahalia at ¶ 0117, “[t]o close a virtual channel, a message packet is transmitted”).

The Examiner does not indicate and the Appellant does not discern any part of E that teaches the above aspects of Claim 12. Therefore, the Examiner has failed to establish that E and Vahalia, alone or in combination, teach all the limitations of Claim 12. Accordingly, this claim is separately patentable and the Appellant respectfully requests that the obviousness rejection be reversed.

4. Claim 13

a) Claim 13 depends from patentable Claim 12

Claim 13 depends from patentable Claim 12. Thus, Claim 13 is directed toward allowable subject matter for at least the reasons mentioned in connection with Claim 12. The Examiner has not relied upon and the Appellant is unable to discern any part of Vahalia that cures the aforementioned deficiencies regarding the claim limitations of Claims 12. Accordingly, for at least these reasons, the Appellant respectfully requests that the § 103(a) rejection of Claim 13 be overturned. Further, the Appellant believes that this claim is separately patentable for the reasons below.

b) Claim 13 is not obvious at least because E and Vahalia, alone or in combination, fail to teach or suggest a persistency handler

Claim 13 recites: “a persistency handler.” E and Vahalia, alone or in combination, do not teach or suggest this additional aspect.

The Examiner contends that Vahalia discloses a persistency handler at ¶ 0123-0125 (*See* Final Office Action mailed June 7, 2007 at pg. 6). The Appellant respectfully disagrees.

As described above, Vahalia discloses “stream handler routines” that manage the transmission of message packets between a data mover and a network client, as well as between two data movers in ¶ 0123-0125. However, a handler that manages message packet transmission does not teach or suggest a persistency handler that maintains the consistency of the

configuration because Vahalia's handlers do not maintain that the data throughout changes is persistently updated, but rather transmit messages upon command between a data mover and a network client or between two data movers.

The Examiner does not indicate and the Appellant does not discern any part of E that teaches the above aspect of Claim 13. Therefore, the Examiner has failed to establish that E and Vahalia, alone or in combination, teach all the limitations of Claim 13. Accordingly, this claim is separately patentable and the Appellant respectfully requests that the obviousness rejection be reversed.

5. **Claim 14**

a) Claim 14 depends from patentable Base Claim 11

Claim 14 depends from patentable Base Claim 11. Thus, Claim 14 is directed toward allowable subject matter for at least the reasons mentioned in connection with Independent Claim 11. The Examiner has not relied upon and the Appellant is unable to discern any part of Vahalia that cures the aforementioned deficiencies regarding the claim limitations of Independent Claims 11. Accordingly, for at least these reasons, the Appellant respectfully requests that the § 103(a) rejection of Claim 14 be overturned. Further, the Appellant believes that this claim is separately patentable for the reasons below.

b) Claim 14 is not obvious at least because E and Vahalia, alone or in combination, fail to teach or suggest a configuration handler to permit access to and modification of the configuration

Claim 14 recites: "a configuration handler to permit access to and modification of the configuration." E and Vahalia, alone or in combination, do not teach or suggest this additional aspect.

The Examiner contends that Vahalia discloses a consistency handler at ¶ 0123-0125 (See Final Office Action mailed June 7, 2007 at pg. 6). The Appellant respectfully disagrees.

As described above, Vahalia discloses "stream handler routines" that manage the transmission of message packets between a data mover and a network client, as well as between

two data movers in ¶ 0123-0125. However, the handlers that manage message packets fails to include the aspect of “a configuration handler to permit access to and modification of the configuration.” The message transmission of the cited reference that prepares messages for transmission is silent with respect to access and modification of a configuration.

The Examiner does not indicate and the Appellant does not discern any part of E that teaches the above aspect of Claim 14. Therefore, the Examiner has failed to establish that E and Vahalia, alone or in combination, teach all the limitations of Claim 14. Accordingly, this claim is separately patentable and the Appellant respectfully requests that the obviousness rejection be reversed.

6. Claim 27

a) Claim 27 depends from patentable Base Claim 25

Claim 27 depends from patentable Base Claim 25. Thus, Claim 27 is directed toward allowable subject matter for at least the reasons mentioned in connection with Independent Claim 25. The Examiner has not relied upon and the Appellant is unable to discern any part of Vahalia that cures the aforementioned deficiencies regarding the claim limitations of Independent Claims 25. Accordingly, for at least these reasons, the Appellant respectfully requests that the § 103(a) rejection of Claim 27 be overturned. Further, the Appellant believes that this claim is separately patentable for the reasons below.

b) Claim 27 is not obvious at least because E and Vahalia, alone or in combination, fail to teach or suggest a configuration cache and a configuration handler resident in each node of the distributed environment

Claim 27 recites: “a configuration cache resident in each node of the distributed environment; and a configuration handler resident in each node of the distributed environment.” E and Vahalia, alone or in combination, do not teach or suggest these additional aspects.

The Examiner contends that ¶ 0123-0125 and Fig. 18, elements 330, 323, and 324 of Vahalia also describe these aspects (*See* Final Office Action mailed June 7, 2007 at pg. 6). The Appellant respectfully disagrees with this contention.

There is no indication that Vahalia's "stream handler routines" that manage the transmission of message packets exist "in each node of the distributed environment." The cited reference is silent with respect to a cache and handler resident in **each** node. Vahalia fails to discuss nodes altogether. As described above, elements 330, 323, and 324 of Figure 18 identify a cached disk array, a data mover #2, and a data mover #1, respectively. The data movers represent the two possible paths connecting clients to the files. However, these paths and a cached disk array do not expressly or inherently describe nodes and there is no identification that any type of cache or handler would be resident inside each of the "nodes."

The Examiner does not indicate and the Appellant does not discern any part of E that teaches the above aspects of Claim 27. Therefore, the Examiner has failed to establish that E and Vahalia, alone or in combination, teach all the limitations of Claim 27. Accordingly, this claim is separately patentable and the Appellant respectfully requests that the obviousness rejection be reversed.

7. **Claims 9, 15, and 26**

Claims 9, 15, and 26 stand rejected under 35 U.S.C. § 103(a) as being obvious over E, in view of the Appellant's alleged admission of prior art.

a) Claim 9 depends from patentable Claim 2, Claim 15 depends from patentable Base Claim 11, and Claim 26 depends from patentable Base Claim 25

Claim 9 depends from patentable Claim 12, Claim 15 depends from patentable Base Claim 11, and Claim 26 depends from patentable Base Claim 25. Therefore, Claims 9, 15, and 26 are directed toward allowable subject matter for at least the reasons mentioned in connection with Claims 2, 11, and 25. The Examiner has not relied upon and the Appellant is unable to discern any part of the Appellant's alleged admission of prior art that cures the aforementioned deficiencies regarding the claim limitations of Claims 2, 11, and 25. Accordingly, for at least

these reasons, the Appellant respectfully requests that the § 103(a) rejection of Claims 9, 15, and 26 be reversed. Further, the Appellant believes that these claims are separately patentable for the reasons below.

b) Claims 9, 15, and 26 are not obvious at least because E, Vahalia, and the Appellant's Background, alone or in combination, fail to teach or suggest Java 2 Enterprise Edition (J2EE) as applied to the claims

Claim 9 recites: “sending a message to a plurality of Java 2 Enterprise Edition (J2EE) nodes,” Claim 15 recites: “a plurality of persistent objects holding information about a Java 2 enterprise edition cluster,” and Claim 26 recites: “a plurality of persistent objects holding information about a Java 2 Enterprise Edition cluster.”

The Examiner contends that ¶ 0007 of the Appellant’s Background in the Specification renders it obvious to one of ordinary skill in the art with respect to the use of J2EE. The Appellant respectfully avers to the contrary.

¶ 0007 of the Appellant’s Background in the Specification indicates the general features of a J2EE environment. The Background describes a J2EE environment having a business layer, a presentation layer, and a user interface layer. The Appellant’s inclusion of such description does not teach or suggest a specific application of J2EE in the context of the claims with respect to nodes and holding such J2EE information in persistent objects.

Therefore, the Examiner has failed to establish that E and Vahalia, alone or in combination, teach all the limitations of claims 9, 15, and 26. Accordingly, these claims are separately patentable and the Appellant respectfully requests that the obviousness rejection be reversed.

8. Claim 16

Claim 16 stands rejected under 35 U.S.C. § 103(a) as being obvious over E, in view of Vahalia, and further in view of the Appellant’s alleged admission of prior art. Claim 16 depends from patentable Claim 15 and is directed toward allowable subject matter for at least the reasons

mentioned in connection with Claim 15. The Examiner has not relied upon and the Appellant is unable to discern any part of Vahalia and the Appellant's alleged admission of prior art that cures the aforementioned deficiencies regarding the claim limitations of Claim 15. Accordingly, for at least these reasons, the Appellant respectfully requests that the § 103(a) rejection of Claim 16 be overturned.

In view of the foregoing, it is respectfully requested that the rejection under 35 U.S.C. § 103(a) be reversed.

For the reasons set forth above, the Appellant respectfully requests the Board overturn the rejection of claims 1-27.

Respectfully submitted,

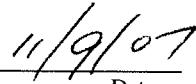
BLAKELY, SOKOLOFF, TAYLOR, & ZAFMAN, LLP

Dated: 11/09/2007

____/Thomas M. Coester/____
Thomas M. Coester, Reg. No. 39,637

1279 Oakmead Parkway
Sunnyvale, California 94085-4040
Telephone (310) 207-3800
Facsimile (310) 820-5988

CERTIFICATE OF TRANSMISSION
I hereby certify that this correspondence is being submitted electronically
via EFS Web on the date shown below to the United States Patent and
Trademark Office.

Lori Ciccio 11/9/07

Date

VIII. CLAIMS APPENDIX

The claims involved in this Appeal are as follows:

1. (Original) A method comprising:
storing a configuration for a distributed environment in a central storage of the distributed environment; and
updating a portion of the configuration in the distributed environment.
2. (Original) The method of Claim 1 wherein updating comprises:
acquiring a lock for the portion of the configuration in a first node in the distributed environment;
modifying the portion of the configuration;
invalidating a representation of the portion of the configuration in the distributed environment; and
releasing the lock.
3. (Original) The method of Claim 2 wherein updating further comprises:
updating a database to reflect modifications of a portion of the configuration; and
blocking reads of the configuration during the updating.
4. (Original) The method of Claim 2 wherein updating further comprises:
notifying nodes in the distributed environment of the updated configuration data.
5. (Original) The method of Claim 2 wherein the lock is cluster wide.
6. (Original) The method of Claim 2 wherein updating further comprises:
writing changes to a shared database.
7. (Original) The method of Claim 2 wherein modifying comprises:
changing a configuration object in a branch of a tree structure.
8. (Original) The method of Claim 2 wherein invalidating comprises:
sending a cache invalidation event to another node in the cluster.

9. (Original) The method of Claim 2 wherein invalidating comprises: sending a message to a plurality of Java 2 Enterprise Edition (J2EE) nodes.
10. (Original) The method of Claim 2 wherein updating further comprises: notifying registered listeners that the configuration has been changed.
11. (Original) A system comprising:
 - a plurality of nodes each having a instance of a configuration manager to maintain consistent storage of a configuration across the nodes without passing configuration modifications between the nodes;
 - a locking server shared by the plurality of nodes to coordinate access to the configuration; and
 - a database management system to provide an interface with a shared relational database, the database to store the configuration.
12. (Original) The system of Claim 11 wherein the configuring manager comprises:
 - a configuration cache; and
 - a configuration handler.
13. (Original) The system of Claim 12 wherein the configuration manager further comprises:
 - a persistency handler.
14. (Original) The system of Claim 11 further comprising:
 - a configuration handler to permit access to and modification of the configuration.
15. (Original) The system of Claim 11 wherein the configuration comprises:
 - a plurality of persistent objects holding information about a Java 2 enterprise edition cluster.
16. (Original) The system of Claim 15 wherein some of the persistent objects are cacheable.
17. (Original) The system of Claim 11 wherein the configuration manager comprises:
 - a change event listener to notify registered components of configuration change events.

18. (Original) A computer readable storage media containing executable computer program instructions which when executed cause a digital processing system to perform a method comprising:

storing a configuration for a distributed environment in a central storage of the distributed environment; and

updating a portion of the configuration in the distributed environment.

19. (Original) The computer readable storage media of Claim 18 containing executable computer program instructions which when executed cause a digital processing system to perform the method wherein updating comprises:

acquiring a lock for the portion of the configuration in a first node in the distributed environment;

modifying the portion of the configuration;

invalidating a representation of the portion of the configuration in the distributed environment; and

releasing the lock.

20. (Original) The computer readable storage media of Claim 19 containing executable computer program instructions which when executed cause a digital processing system to perform the method wherein updating comprises:

updating a database to reflect modifications of a portion of the configuration; and

blocking reads of the configuration during the updating.

21. (Original) The computer readable storage media of Claim 19 containing executable computer program instructions which when executed cause a digital processing system to perform the method wherein updating comprises:

notifying node in the distributed environment of the current configuration data.

22. (Original) The computer readable storage media of Claim 19 containing executable computer program instructions which when executed cause a digital processing system to perform the method wherein updating further comprises:

changing the configuration locally;

writing the changes to a shared database; and
committing the changes.

23. (Original) The computer readable storage media of Claim 19 containing executable computer program instructions which when executed cause a digital processing system to perform the method wherein invalidating comprises:

 sending a cache invalidation event to another node in the cluster.

24. (Original) The computer readable storage media of Claim 19 containing executable computer program instructions which when executed cause a digital processing system to perform the method wherein updating comprises:

 notifying registered listeners that the configuration has been changed.

25. (Original) A system comprising:

 means for maintaining consistent storage of configuration information in a distributed environment;

 means for controlling access to the configuration information; and

 means for interfacing with a relational database system to provide persistent storage of the configuration information.

26. (Original) The system of Claim 25 wherein the configuration information comprises:
 a plurality of persistent objects holding information about a Java 2 Enterprise Edition cluster.

27. (Original) The system of Claim 25 wherein the means for maintaining comprises:
 a configuration cache resident in each node of the distributed environment; and
 a configuration handler resident in each node of the distributed environment.

IX. EVIDENCE APPENDIX

No evidence is submitted with this appeal.

X. RELATED PROCEEDINGS APPENDIX

No related proceedings exist.